Algorithmica (2009) 54: 226–242
DOI 10.1007/s00453-007-9136-0

Faster Deterministic Communication in Radio Networks

Ferdinando Cicalese · Fredrik Manne · Qin Xin

Received: 22 January 2007 / Accepted: 20 November 2007 / Published online: 4 December 2007

Abstract We study the communication primitives of broadcasting (one-to-all communication) and gossiping (all-to-all communication) in known topology radio networks, i.e., where for each primitive the schedule of transmissions is precomputed based on full knowledge about the size and the topology of the network. We show that gossiping can be completed in $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$ time units in any radio network of size n, diameter D, and maximum degree $\Delta = \Omega(\log n)$. This is an almost optimal schedule in the sense that there exists a radio network topology, specifically a Δ-regular tree, in which the radio gossiping cannot be completed in less than $\Omega(D + \frac{\Delta \log n}{\log \Delta})$ units of time. Moreover, we show a $D + O(\frac{\log^3 n}{\log\log n})$ schedule for the broadcast task. Both our transmission schemes significantly improve upon the currently best known schedules by Gąsieniec, Peleg, and Xin (Proceedings of the 24th Annual ACM SIGACT-SIGOPS PODC, pp. 129–137, 2005), i.e., a $O(D + \Delta \log n)$ time schedule for gossiping and a $D + O(\log^3 n)$ time schedule for broadcast. Our broadcasting schedule also improves, for large D, a very recent $O(D + \log^2 n)$ time broadcasting schedule by Kowalski and Pelc.

A preliminary version of this paper appeared in the proceedings of ISAAC'06.
F. Cicalese supported by the Sofja Kovalevskaja Award 2004 of the Alexander von Humboldt Stiftung. F. Manne and Q. Xin supported by the Research Council of Norway through the SPECTRUM project.

F. Cicalese
AG Genominformatik, Technische Fakultät, Universität Bielefeld, Bielefeld, Germany
e-mail: nando@cebitec.uni-bielefeld.de

F. Manne
Department of Informatics, The University of Bergen, P.B. 7800, Bergen 5020, Norway
e-mail: fredrikm@ii.uib.no

Q. Xin (✉)
Simula Research Laboratory, P.B. 134, Lysaker 1325, Norway
e-mail: xin@simula.no

Keywords Centralized radio networks · Broadcasting · Gossiping

1 Introduction

We consider the following model of a radio network: an undirected connected graph $G = (V, E)$, where V represents the set of nodes of the network and E contains unordered pairs of distinct nodes, such that $(v, w) \in E$ iff the transmissions of node v can directly reach node w and vice versa (the reachability of transmissions is assumed to be a symmetric relation). In this case, we say that the nodes v and w are *neighbours* in G. Note that in a radio network, a message transmitted by a node is always sent to all of its neighbours.

The *degree* of a node is the number of its neighbours. We use Δ to denote the *maximum degree* of the network, i.e., the maximum degree of any node in the network. The *size of the network* is the number of nodes $n = |V|$.

Communication in the network is synchronous and consists of a sequence of communication steps. In each step, a node v either transmits or listens. If v transmits, then the transmitted message reaches each of its neighbours by the end of this step. However, a node w adjacent to v successfully receives this message iff in this step w is listening and v is the only transmitting node among w's neighbours. If node w is adjacent to a transmitting node but it is not listening, or it is adjacent to more than one transmitting node, then a *collision* occurs and w does not retrieve any message in this step.

The two classical problems of information dissemination in computer networks are the *broadcasting* problem and the *gossiping* problem. The broadcasting problem requires distributing a particular message from a distinguished *source* node to all other nodes in the network. In the gossiping problem, each node v in the network initially holds a message m_v, and the aim is to distribute all messages to all nodes. For both problems, one generally considers as the efficiency criterion the minimization of the time needed to complete the task.

In the model considered here, the running time of a communication schedule is determined by the number of time steps required to complete the communication task. This means that we do not account for any internal computation within individual nodes. Moreover, no limit is placed on the length of a message that one node can transmit in one step. In particular, this assumption plays an important role in the case of the gossiping problem, where it is then assumed that in each step when a node transmits, it transmits all the messages it has collected by that time. (i.e., the ones received so far and its own one.)

Our schemes rely on the assumption that the communication algorithm can use complete information about the network topology. Such topology-based communication algorithms are useful whenever the underlying radio network has a fairly stable topology/infrastructure. As long as no changes occur in the network topology during the execution of the algorithm, the tasks of broadcasting and gossiping will be completed successfully. Here, we shall not touch upon reliability issues. However, we remark that it is possible to increase the level of fault-tolerance in our algorithms, at the expense of some small extra time consumption.

Our results We provide a new (efficiently computable) deterministic schedule that uses $O(D+\frac{\Delta\log n}{\log\Delta-\log\log n})$ time units to complete the gossiping task in any radio network of size n, diameter D and maximum degree $\Delta=\Omega(\log n)$. This significantly improves on the previously known best schedule, i.e., the $O(D+\Delta\log n)$ schedule of [7]. Remarkably, our new gossiping scheme constitutes an almost optimal schedule in the sense that there exists a radio network topology, specifically a Δ-regular tree, in which the radio gossiping cannot be completed in less than $\Omega(D+\frac{\Delta\log n}{\log\Delta})$ units of time.

For the broadcast task, we show a new (efficiently computable) radio schedule that works in time $D+O(\frac{\log^3 n}{\log\log n})$, improving the currently best published result for arbitrary topology radio networks, i.e., the $D+O(\log^3 n)$ time schedule proposed by Gąsieniec et al. in [7]. It is noticeable that for large D, our scheme also outperforms the very recent (asymptotically optimal) $O(D+\log^2 n)$ time broadcasting schedule by Kowalski and Pelc in [10]. This is because of the significantly larger coefficient of the D term hidden in the asymptotic notation. In fact, in our case the D term comes with coefficient 1.

Related work The work on communication in known topology radio networks was initiated in the context of the broadcasting problem. In [3], Chlamtac and Weinstein prove that the broadcasting task can be completed in time $O(D\log^2 n)$ for every n-vertex radio network of diameter D. An $\Omega(\log^2 n)$ time lower bound was proved for the family of graphs of radius 2 by Alon et al [1]. In [5], Elkin and Kortsarz give an efficient deterministic construction of a broadcasting schedule of length $D+O(\log^4 n)$ together with a $D+O(\log^3 n)$ schedule for planar graphs. Recently, Gąsieniec, Peleg and Xin [7] showed that a $D+O(\log^3 n)$ schedule exists for the broadcast task, that works in *any* radio network. In the same paper, the authors also provide an optimal randomized broadcasting schedule of length $D+O(\log^2 n)$ and a new broadcasting schedule using fewer than $3D$ time slots on planar graphs. A $D+O(\log n)$-time broadcasting schedule for planar graphs has been showed in [11] by Manne, Wang, and Xin. Very recently, a $O(D+\log^2 n)$ time deterministic broadcasting schedule for any radio network was proposed by Kowalski and Pelc in [10]. This is asymptotically optimal unless $NP\subseteq BPTIME(n^{\mathcal{O}(\log\log n)})$ [10]. Nonetheless, for large D, our $D+O(\frac{\log^3 n}{\log\log n})$ time broadcasting scheme outperforms the one in [10], because of the larger coefficient of the D term hidden in the asymptotic notation describing the time evaluation of this latter scheme.

Efficient radio broadcasting algorithms for several special types of network topologies can be found in [4]. For general networks, however, it is known that the computation of an optimal (radio) broadcast schedule is NP-hard, even if the underlying graph is embedded in the plane [2, 13].

Radio gossiping in networks with known topology was first studied in the context of radio communication with messages of limited size, by Gąsieniec and Potapov in [8]. They proposed several optimal or close to optimal $O(n)$-time gossiping procedures for various standard network topologies, including lines, rings, stars and free trees. In the same paper, an $O(n\log^2 n)$ gossiping scheme for general topology radio network is provided and it is proved that there exists a radio network topology in which the gossiping (with unit size messages) requires $\Omega(n\log n)$ time. In [12], Manne and Xin show the optimality of this bound by providing an $O(n \log n)$-time gossiping schedule with unit size messages in any radio network. The first work on radio gossiping in known topology networks with arbitrarily large messages is [9], where several optimal gossiping schedules are shown for a wide range of radio network topologies. For arbitrary topology radio networks, an $O(D + \Delta \log n)$ schedule was given by Gąsieniec, Peleg, and Xin in [7]. To the best of our knowledge no better result is known to date for arbitrary topology.

2 Gossiping in General Graphs with Known Topology

The gossiping task can be performed in two consecutive phases. During the first phase we gather all individual messages in one (central) point of the graph. Then, during the second phase, the collection of individual messages is broadcast to all nodes in the network.

We start this section with the presentation of a simple gathering procedure that works in time $O((D + \Delta)\frac{\log n}{\log \Delta - \log\log n})$ in free trees. Later we show how to choose a spanning breadth-first (BFS) tree in an arbitrary graph G in order to gather (along its branches) all messages in G also in time $O((D + \Delta)\frac{\log n}{\log \Delta - \log\log n})$, despite the additional edges in G which might potentially cause collisions. Finally, we show how the gathering process can be pipelined and sped up to run in $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$ time.

2.1 A Super-ranking Procedure

Given an arbitrary tree, we choose its central node c as the root. Then, the nodes in the tree (rooted at c) are partitioned into consecutive layers $L_i = \{v \mid dist(c, v) = i\}$, for $i = 0, \ldots, r$ where r is a radius of the tree. We denote the size of each layer L_i by $|L_i|$.

We use a non-standard approach for ranking the nodes in a rooted tree, which we call *super-ranking*. The super-ranking depends on an integer parameter $2 \le x \le \Delta$, that for our purposes will be optimized later. Specifically, for every leaf v we define $\mathsf{rank}(v, x) = 1$. Then, for a non-leaf node, v with children $v_1, \ldots, v_k$, we define $\mathsf{rank}(v, x)$ as follows. Let $\hat{r} = \max_{i=1,\ldots,k}\{\mathsf{rank}(v_i, x)\}$. If at least x of the children of v have rank $\hat{r}$, then $\mathsf{rank}(v, x) = \hat{r} + 1$ otherwise $\mathsf{rank}(v, x) = \hat{r}$.

For an example, see Fig. 1, where the same tree is ranked with threshold $x = 3$ and $x = 2$ respectively.

For each $x \ge 2$, we define $r^{[x]}_{\max} = \max_{v \in T} \mathsf{rank}(v, x)$. As an immediate consequence of the definition of $\mathsf{rank}(\cdot, \cdot)$ we have the following.

Lemma 1 *Let T be a tree with n nodes of maximum degree Δ. Then, $r^{[x]}_{\max} \le \lceil \log_x n \rceil$, for each $2 \le x \le \Delta$.*

Note that when $x = 2$ we obtain the standard ranking procedure, that has been employed in the context of radio communication in known topology networks in [6, 7, 9]. Previously this same ranking had been used to define the *Strahler number* of

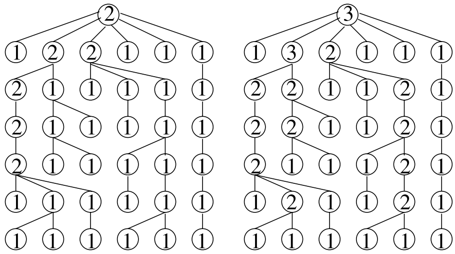

Fig. 1 A tree of size $n = 37$ ranked with $x = 3$ (*left*) and $x = 2$ (*right*)

binary trees, introduced in hydrogeology [14] and extensively studied in computer science (cf. [15] and the references therein).

The schedule for gathering messages at the root is now defined in stages using the super-ranked tree under the assumption that the value of the parameter x has been fixed. For the sake of the analysis, we will optimize its value later. We partition the nodes of the tree into different *rank sets* that are meant to separate the stages in which nodes are transmitting, i.e., nodes from different rank sets transmit in different stages.

For $x \geq 2$, let $r_{\max}^{[x]}$ be the maximum rank for a node of T according to the super-ranking with parameter x. Recall that $r_{\max}^{[x]} \leq \lceil \log_x n \rceil$. Then, let $R_i(x) = \{v \mid \mathsf{rank}(v, x) = i\}$, where $1 \leq i \leq r_{\max}^{[x]}$.

We use the above rank sets to partition the node set. In the following, we shall use two rankings of the nodes: one with the parameter x set to some (fixed) value greater than 2 and one with $x = 2$.

Definition 1 The fast transmission set is given by $F_j^k = \{v \mid v \in L_k \cap R_j(2)$ and $parent(v) \in R_j(2)\}$. Also define $F_j = \bigcup_{k=1}^{D} F_j^k$ and $F = \bigcup_{j=1}^{r_{\max}^{[2]}} F_j$.

Definition 2 The slow transmission set is given by $S_j^k = \{v \mid v \in L_k \cap R_j(2)$ and $parent(v) \in R_p(2)$, for some $p > j$; and $\mathsf{rank}(v, x) = \mathsf{rank}(parent(v), x), x > 2\}$. Also define $S_j = \bigcup_{k=1}^{D} S_j^k$ and $S = \bigcup_{j=1}^{r_{\max}^{[2]}} S_j$.

Definition 3 The *super-slow transmission set* is given by $SS_j^k = \{v \mid v \in L_k \cap R_j(x)$ and $parent(v) \in R_i(x), i > j\}$. Accordingly, define $SS_j = \bigcup_{k=1}^{D} SS_j^k$ and $SS = \bigcup_{j=1}^{r_{\max}^{[x]}} SS_j$.

Note that the above transmission sets define a partition of the node set. Each node v only belongs to one of the transmission sets and $V = F \cup S \cup SS$.

Lemma 2 *Fix positive integers $i \leq r_{\max}^{[x]}$, $j \leq r_{\max}^{[2]}$ and $k \leq D$. Then, during the ith stage, all nodes in F_j^k can transmit to their parents simultaneously without any collisions.*

Proof Consider any two distinct nodes u and v in F_j^k, and suppose they interfere with each other. This is true if they have a common neighbour in L_{k-1}. Obviously, u and v are on the same level and must therefore have the same parent y in the tree. Moreover, according to the definition of the fast transmission set F_j^k, $u, v, y \in R_j(2)$. However, according to the definition of the super-ranking procedure, if $\mathsf{rank}(u, 2) = \mathsf{rank}(v, 2) = j$ then $\mathsf{rank}(y, 2)$ must be at least $j + 1$. Hence the nodes u and v cannot both belong to F_j^k, which leads to a contradiction. □

Lemma 3 *Fix positive integers $i \le r_{\max}^{[x]}$, $j \le r_{\max}^{[2]}$ and $k \le D$. Then, all messages from nodes in $S_j^k \cap R_i(x)$ can be gathered in their parents in at most $x - 1$ time units.*

Proof By Definition 1 we have that for each node v in $S_j^k \cap R_i(x)$, *parent*(v) has at most $x - 1$ children in $S_j^k \cap R_i(x)$, for each $i = 1, 2, \dots, r_{\max}^{[x]} \le \lceil \log_x n \rceil$, $j = 1, 2, \dots, r_{\max}^{[2]} \le \lceil \log n \rceil$ and $k = 1, \dots, D$. Therefore, the desired result is achieved by simply letting each parent of nodes in $S_j^k \cap R_i(x)$ collect the messages from its children one at a time. □

We shall use the following result from [7].

Proposition 1 [7] *There exists a gathering procedure Γ such that in any graph G of maximum degree Δ_G and diameter D_G the gossiping task, and in particular the gathering stage, can be completed in time $O(D_G + \Delta_G \log n)$.*

The following procedure moves messages from all nodes v with $\mathsf{rank}(v, x) = i$ into their lowest ancestor u with $\mathsf{rank}(u, x) \ge i + 1$, where $x > 2$, using the gathering procedure Γ from the previous proposition.

Procedure SUPER-GATHERING(i)

1. *Move messages from nodes in* $(F \cup S) \cap R_i(x)$ to SS_i;
   using the gathering procedure Γ in Proposition 1.
2. *Move messages from nodes in SS_i to their parents*;
   all parents collect their messages from their children in SS_i one by one.

Note that the subtrees induced by the nodes in $R_i(x)$ have maximum degree $\le x$. Thus, by Proposition 1 and Lemma 3, we have that the time complexity of step 1 is $O(D + x \log n)$. The time complexity of step 2 is bounded by $O(\Delta)$, where Δ is the maximum degree of the tree. By Lemma 1, $r_{\max}^{[x]} \le \lceil \log_x n \rceil$. Thus, we have that the procedure SUPER-GATHERING completes the gathering stage in time $O((D + \Delta + x \log n) \log_x n)$. Since we can have this followed by the trivial broadcasting stage performed in time $O(D)$, we have proved the following.

Theorem 1 *In any tree of size n, diameter D and maximum degree Δ, the gossiping task can be completed in time $O((D + \Delta + x \log n) \log_x n)$, where $2 < x \le \Delta$. In particular when $\Delta = \Omega(\log n)$, by choosing $x = \frac{\Delta}{\log n}$, we obtain the bound $O((D + \Delta) \frac{\log n}{\log \Delta - \log\log n})$.*

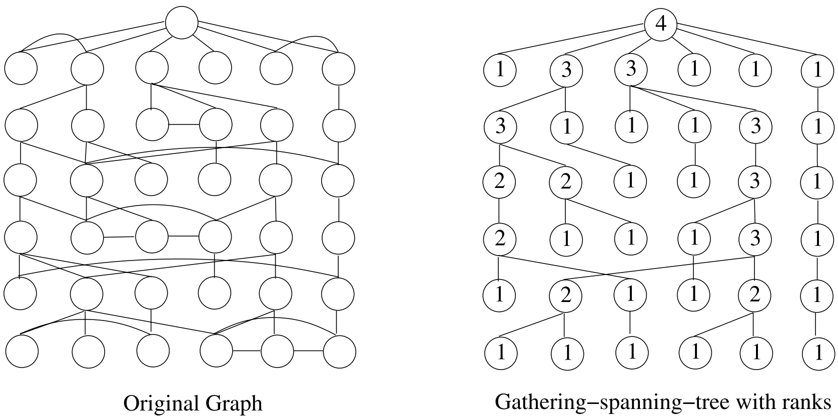

Fig. 2 Creating a gathering spanning tree

2.2 Gathering Messages in Arbitrary Graphs

We start this section with the introduction of the novel concept of a *super-gathering spanning tree* (*SGST*). Such tree plays a crucial role in our gossiping-scheme for arbitrary graphs. We shall show an $O(n^2 \log n)$-time algorithm that constructs a *SGST* in an arbitrary graph G of size n and diameter D. In the concluding part of this section, we propose a new more efficient schedule that completes message gathering in time $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$.

A *super-gathering spanning tree* (*SGST*) for a graph $G = (V, E)$ is any BFS spanning tree T_G of G, ranked according to the super-ranking above[1] and satisfying:

(1) T_G is rooted at the central node c of G;
(2) T_G is ranked;
(3) all nodes in F_j^k of T_G are able to transmit their messages to their parents simultaneously without any collision, for all $1 \le k \le D$ and $1 \le j \le r_{\max}^{[2]} \le \lceil \log n \rceil$;
(4) every node v in $S_j^k \cap R_i(x)$ of T_G has following property: *parent*(v) has at most $x - 1$ neighbours in $S_j^k \cap R_i(x)$, for all $i = 1, 2, \ldots, r_{\max}^{[x]} \le \lceil \log_x n \rceil$, $j = 1, 2, \ldots, r_{\max}^{[2]} \le \lceil \log n \rceil$ and $k = 1, \ldots, D$.

Any BFS spanning tree T_G of G satisfying only conditions (1), 2), and (3) above is called a *gathering spanning tree*, or simply *GST*. Figure 2 shows an example of a *GST*.

We recall the following result from [7].

Theorem 2 *There exists an efficient* ($O(n^2 \log n)$ *time*) *construction of a* GST *on an arbitrary graph* G (*see Theorem* 2.5 *in* [7]).

[1] We use Definitions 1–3 of the ranking partitions given above.

```
Procedure SUPER-GATHERING-SPANNING-TREE(GST)
 (1) Fix rank(w, 2) for every node w ∈ V;
 (2) For k := D down to 1 do
 (3)     For i := r_max^[x] down to 1 do
 (4)         For j := r_max^[2] down to 1 do
 (5)             While ∃v ∈ S_j^k ∩ R_i(x) in GST such that
                 |S_j^k ∩ R_i(x) ∩ NB(parent(v))| ≥ x
             do
 (6)                 rank(parent(v), x) = i + 1;  //rank(v, x) = i
 (7)                 UPDATE = {u|u ∈ S_j^k ∩ R_i(x) ∩ NB(parent(v))};
 (8)                 SS^k_rank(v,x) = SS^k_rank(v,x) ∪ UPDATE;
 (9)                 E_GST = E_GST − {(u, parent(u))|u ∈ UPDATE};
(10)                 E_GST = E_GST ∪ {(u, parent(v))|u ∈ UPDATE};
(11)                 S_j^k = S_j^k − {u|u ∈ UPDATE};
(12)                 re-set rank(w, x) for each w ∈ V;
(13)                 recompute the sets S and SS in GST
```

Fig. 3 Procedure Super-Gathering-Spanning-Tree

The procedure SUPER-GATHERING-SPANNING-TREE, presented next, in Fig. 3, constructs a super-gathering-spanning-tree $SGST \subseteq G$ on the basis of a $GST \subseteq G$ using a *pruning process*. The pruning process is performed layer by layer starting from the bottom (layer D) of the GST. For each layer we gradually *fix* the parents of all nodes which violate condition (4) above, i.e., each v in $S_j^k \cap R_i(x)$ of GST, such that $parent(v)$ has at least x neighbours in $S_j^k \cap R_i(x)$. In fact, for our gathering-scheme, v is a node which is potentially involved in collisions. In each layer, the pruning process starts with the nodes of highest rank in the current layer. We use $NB(v)$ to denote the set of neighbours of the node v in the original graph G. In Fig. 4, we show the output of the SUPER-GATHERING-SPANNING-TREE procedure when it is run on the GST presented in Fig. 2.

We prove that Procedure SUPER-GATHERING-SPANNING-TREE constructs the $SGST$ of an arbitrary graph $G = (V, E)$ in time $O(n^2 \log n)$. The following technical lemma is easily proved by induction.

Lemma 4 *After completing the pruning process at layer k in GST, the structure of edges in GST between layers $k-1, \dots, D$ is fixed, i.e., each node v within layers $k, \dots, D$ in all sets $S_j^k \cap R_i(x)$, satisfy the following property: $parent(v)$ has at most $x-1$ neighbours in $S_j^k \cap R_i(x)$, for $i = 1, \dots, r_{\max}^{[x]} \le \lceil \log_x n \rceil$ and $j = 1, \dots, r_{\max}^{[2]} \le \lceil \log n \rceil$.*

Proof We rely on the assumption that before the kth execution of the outer loop, all edges in GST between layers from k through D are already fixed and will never change again. Note that during the pruning process at layer k, the updates involve only some edges between layers k and $k-1$. Note also that the updates at layer k are always executed first at the nodes with higher ranks. Thus, when a node v with the

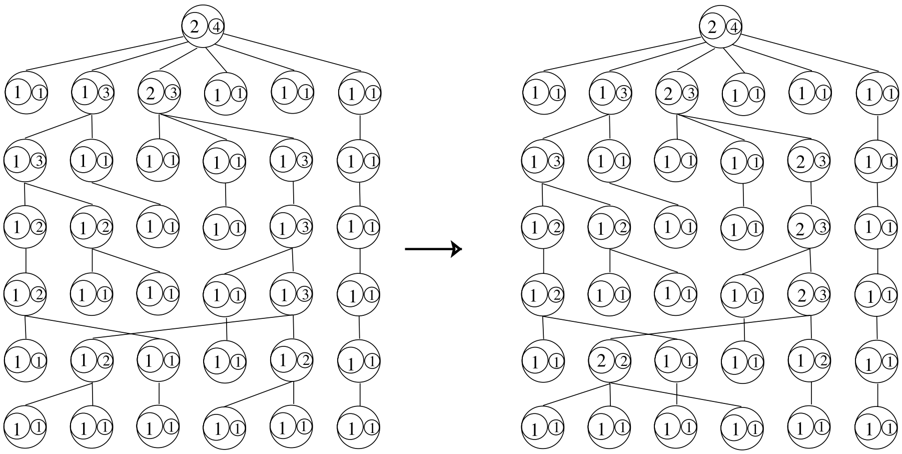

Fig. 4 From gathering-spanning-tree to super-gathering-spanning-tree

property that *parent*(v) has at least x neighbours in $S_j^k \cap R_i(x)$ is found, v gets pruned. The nodes in $S_j^k \cap R_i(x)$ that are neighbours of *parent*(v) are assigned *parent*(v) as their parent and are moved to the super-slow transmission set. From now on, neither the edges $(u, parent(u))$, $u \in S_j^k \cap R_i(x)$ nor the edges leading to their new parent *parent*(v) will be considered again. This is because these nodes $u \in S_j^k \cap R_i(x)$ no longer belong to the set S and further updates at this layer cannot change this property. The lemma follows. □

Lemma 5 *Procedure* SUPER-GATHERING-SPANNING-TREE *preserves the property of the* GST, *in which the transmissions within all sets* F_j^k *are free of collisions*, *for* $j = 1, \ldots, r_{\max} \le \lceil \log n \rceil$ *and* $1 \le k \le D$.

Proof In the procedure SUPER-GATHERING-SPANNING-TREE, the $\mathsf{rank}(v, 2)$ is fixed for each $v \in GST$ and we only update the $\mathsf{rank}(v, x)$ during the pruning process (by moving some nodes in S to SS). Consequently, we did not change the properties of the subtree of the original GST induced by the vertices in F. The lemma follows. □

The above results implies the following theorem.

Theorem 3 *For an arbitrary graph there exists an* $O(n^2 \log n)$ *time construction of a SGST*.

Proof The claim follows directly from Theorem 2, Lemma 4 and the fact that procedure SUPER-GATHERING-SPANNING-TREE preserves the property of the *GST* it starts with. In fact, at the beginning of the SUPER-GATHERING-SPANNING-TREE procedure $\mathsf{rank}(v, 2)$ for each $v \in GST$ is fixed. Then, only $\mathsf{rank}(v, x)$ is updated during the pruning. This does not affect the properties of the subtree of the original GST induced by the vertices in F. During the pruning process, each original edge (v, u)

is only considered once, and also the re-ranking of the *GST* costs at most $O(n)$ time. Consequently, the time to construct a *SGST* is bounded by $O(n^2 \log n)$. □

2.3 An $O((D+\Delta)\frac{\log n}{\log \Delta - \log\log n})$-Time Gossiping Schedule

Using the ranks computed on the *SGST*, the nodes of the graph are partitioned into distinct *rank sets* $R_i = \{v \mid \mathsf{rank}(v,x) = i\}$, where $1 \le i \le r_{\max}^{[x]} \le \lceil \log_x n \rceil$. This allows the gathering of all messages into the central node c, stage by stage, using the structure of the *SGST* as follows. During the ith stage, all messages from nodes in $(F \cup S) \cap R_i(x)$ are first moved to the nodes in SS_i. Later, we move all messages from nodes in SS_i to their parents in *SGST*. In order to avoid collisions between transmissions originating at neighbouring BFS layers we divide the sequence of transmission time slots into three separate (interleaved) subsequences of time slots. Specifically, the nodes in layer L_j transmit in time slot t iff $t \equiv j \pmod 3$.

Lemma 6 *In stage i, the nodes in the set SS_i of the SGST transmit their messages to their parents in time $O(\Delta)$.*

Proof By [9, Lemma 4], one can move all messages between two partitions of a bipartite graph with maximum degree Δ (in this case two consecutive BFS layers) in time Δ. The solution is based on the use of the *minimal covering set*. Note that during this process a (possibly) combined message m sent by a node $v \in SS_i$ may be delivered to the parent of another transmitting node $w \in SS_i$ rather than to *parent*(v). But this is fine, since now the time of delivery of the message m to the root of the tree is controlled by the delivery mechanism of the node w. Obviously this flipping effect can be observed a number of times in various parts of the tree, though each change of the route does not change the delivering mechanism at all.

In order to avoid extra collisions caused by nodes at neighbouring BFS layers, we use the solution with three separate interleaved subsequences of time slots incurring a slowdown with a multiplicative factor of 3. □

When the gathering stage is completed, the gossiping problem is reduced to the broadcasting problem. We distribute all messages to every node in the network by reversing the direction and the time of transmission of the gathering stage. In Sect. 3 we prove that the broadcasting stage can be performed faster in graphs with large Δ, i.e., in time $D + O(\frac{\log^3 n}{\log\log n})$.

Theorem 4 *In any graph G with $\Delta = \Omega(\log n)$, the gossiping task can be completed in time $O((D+\Delta)\frac{\log n}{\log \Delta - \log\log n})$.*

Proof During the ith stage, all messages from $(F \cup S) \cap R_i(x)$ are moved to SS_i. Because of property (4) of the *SGST*, Proposition 1 assures that this can be achieved in time $O(D + x\log n)$. By Lemma 6, all nodes in the set SS_i can transmit their messages to their parents in *SGST* in time $O(\Delta)$. By Lemma 1, this process is repeated at most $\log_x n$ times. Thus, the gossiping time can be bounded by $O((D + \Delta + x\log n)\log_x n)$. The desired result follows directly by setting $x = \frac{\Delta}{\log n}$. □

2.4 The $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$ -Time Gossiping Schedule

The result of Theorem 4 is obtained by a transmission process consisting of $\lceil \log_x n \rceil$ separate stages, each costing $O(D + \Delta + x \log n)$ units of time. We shall now show that the transmissions of different stages can be pipelined and a new gossiping schedule obtained of length $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$.

The communication process will be split into consecutive blocks of 9 time units each. The first 3 units of each block are used for fast transmissions from the set F, the middle 3 units are reserved for slow transmissions from the set S and the remaining 3 are used for super-slow transmissions of nodes from the set SS. We use 3 units of time for each type of transmission in order to prevent collisions between neighbouring BFS layers, like we did in the last section. Recall that we can move all messages between two consecutive BFS layers in time Δ [9, Lemma 4]. Moreover, the same result in [9] together with property (4) of the GSTS, allows us to move all messages stored in $S_j^k \cap R_i(x)$ to their parents in $SGST$ within time $x - 1$.

We compute for each node $v \in S_j \cap R_i(x)$ at layer k the number of a step $1 \leq s(v) \leq x - 1$ in which node v can transmit without interruption from other nodes in $S_j \cap R_i(x)$ also in layer k. We also compute for each node $u \in SS_i$ at layer k the number of a step $1 \leq ss(u) \leq \Delta$ in which the node u can transmit without interruption from other nodes in SS_i also in layer k.

Let v be a node at layer k and with $\mathsf{rank}(v, 2) = j$ and $\mathsf{rank}(v, x) = i$, in $SGST$. Depending on if v belongs to the set F, to the set S or to the set SS, it will transmit in the time block $t(v)$ given by:

$$t(v) = \begin{cases} (D - k + 1) + (j - 1)(x - 1) + (i - 1)\,(\Delta + (x - 1)\log n) & \text{if } v \in F, \\ (D - k + 1) + (j - 1)(x - 1) + s(v) + (i - 1)\,(\Delta + (x - 1)\log n) & \text{if } v \in S, \\ (D - k + 1) + \log n(x - 1) + (i - 1)\,(\Delta + (x - 1)\log n) + ss(v) & \text{if } v \in SS. \end{cases}$$

We observe that any node v in the $SGST$ requires at most D fast transmissions, $\log n$ slow transmissions and $\log_x n$ super-slow transmissions to deliver its message to the root of the $SGST$ if there is no collision during each transmission. Moreover, the above definition of $t(v)$ results in the following lemma.

Lemma 7 *A node v transmits its message as well as all messages collected from its descendants towards its parent in SGST successfully during the time block allocated to it by the transmission pattern.*

Proof Let v be a node at layer k such that $\mathsf{rank}(v, 2) = j$ and $\mathsf{rank}(v, x) = i$. For each node w at layer $k' > k$, which is a descendant of v we have that $\mathsf{rank}(w, 2) = j' \leq j = \mathsf{rank}(v, 2)$ and $\mathsf{rank}(w, x) = i' \leq i = \mathsf{rank}(v, x)$. Therefore if $v, w \in F$, the first term of the expression $(D - k' + 1) + (j' - 1) \cdot (x - 1) + (i' - 1) \cdot (\Delta + (x - 1) \cdot \log n)$ is smaller for w. Hence, according to the pattern of transmissions above, it is not hard to see that node w transmits earlier than node v also holds for other cases (e.g. $v \in SS$ and $w \in F$).

We now prove that any node v following the pattern of transmissions will transmit to its parent without being interrupted by anyone else.

In fact, no collision can happen between neighbouring BFS layers because of the separation into three subsequences, ensuring that three time units are available within each block. Nor can there be collisions between transmissions coming from different transmission-sets (fast, slow and super-slow), because of the three parts of each time block. It remains to rule out collisions between nodes within the same transmission-set and at the same BFS layer in the *SGST*.

Assume that $v, w \in F$ and that they are at the same BFS layer i in the *SGST*.

If v and w also have the same rank $\mathsf{rank}(v,2) = \mathsf{rank}(w,2)$ and $\mathsf{rank}(v,x) = \mathsf{rank}(w,x)$, then they do not interrupt each other due to the properties of *SGST*.

If they have different ranks $\mathsf{rank}(v,2) = j \neq j' = \mathsf{rank}(w,2)$ but $\mathsf{rank}(v,x) = \mathsf{rank}(w,x)$ or $\mathsf{rank}(v,x) = i \neq i' = \mathsf{rank}(w,x)$ but $\mathsf{rank}(v,2) = \mathsf{rank}(w,2)$ respectively, then they transmit in different time blocks according to the pattern of transmissions for the nodes in F.

If $\mathsf{rank}(v,2) = j \neq j' = \mathsf{rank}(w,2)$ and $\mathsf{rank}(v,x) = i \neq i' = \mathsf{rank}(w,x)$, the transmission pattern separates v, w by at least $|(j - j') \cdot (x-1) + (i - i') \cdot (\Delta + (x-1) \cdot \log n)| > \Delta + (x-1))$ time blocks. The inequality follows since $|j - j'| \leq \log n - 1$ and $|i - i'| \leq 1$. Consequently, v and w cannot interfere with each other, either.

Assume now that $v, w \in S$ and that they are at the same BFS layer k in GST.

If $\mathsf{rank}(v,x) = \mathsf{rank}(w,x)$, then either $\mathsf{rank}(v,2) = j \neq j' = \mathsf{rank}(w,2)$ and both $s(v), s(w) \leq (x-1)$, or if they have the same rank j, then they have different values of $s(v)$ and $s(w)$. Hence, they do not interrupt each other.

If $\mathsf{rank}(v,2) = j \neq j' = \mathsf{rank}(w,2)$ and $\mathsf{rank}(v,x) = i \neq i' = \mathsf{rank}(w,x)$, the pattern of transmissions separates v, w by at least $|(j - j') \cdot (x-1) + (s(v) - s(w)) + (i - i') \cdot (\Delta + (x-1) \cdot \log n)| > \Delta$ time blocks. The inequality follows since $|j - j'| \leq \log n - 1$ and $|i - i'| \leq 1$ and $|s(v) - s(w)| \leq x - 2$. Thus, there cannot be a collision between v and w.

Using similar arguments, we can also prove that when $v, w \in SS$ no collision can happen either. This completes the proof. □

Since the number of time blocks used is $\leq D + (x \cdot \log n + \Delta) \cdot (\log_x n + 1)$, we have

Theorem 5 *In any graph G, the gossiping task can be completed in time $O(D + (x \cdot \log n + \Delta) \log_x n)$, where $2 \leq x \leq \Delta$. In particular when $\Delta = \Omega(\log n)$, by setting $x = \frac{\Delta}{\log n}$ the bound becomes $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$.*

By employing the better approximate solution of the equation $\Delta = x \log x$, e.g., taking $x = \frac{\Delta}{\log n - \log\log^\star n}$, we get:

Corollary 1 *In any graph G of $\Delta = \Omega(\log n)$, the gossiping task can be completed in time $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n + \log\log\log^\star n})$.*

Remark 1 Fix integers y_1, y_2, such that $2 \leq y_1 < y_2 \leq \frac{\Delta}{\log n}$. When we use our scheme with $x = y_2$ by Theorem 5 we get the bound $O(D + y_2 \log n + \Delta \log_{y_2} n)$. Suppose now that we tried to resolve all the slow transmissions in $y_1 < y_2$ time slots. This is possible if there are always at most y_1 neighbours in the slow transmissions. Nonetheless, we could force our algorithm to use y_1 slots whenever possible and use the y_2 (sufficient) slots only when there are more than y_1 neighbours to take care of. Obviously, such modified scheme can resort to the y_2 time slots for slow transmissions at most $\log_{y_1} n$ times. Therefore, we have the new time bound $O(D + y_1 \log n + y_2 \log_{y_1} n + \Delta \log_{y_2} n)$.

Of course the same reasoning can be repeated to try and reduce the number of time slots for the slow transmissions that can be accommodated with y_1 time slots. Iterating, we have that for $2 \le x_1 < x_2 < \cdots < x_t \le \frac{\Delta}{\log n}$, there is a gossiping schedule with time complexity

$$O\left(D + x_1 \log n + \Delta \log_{x_t} n + \sum_{i=1}^{t-1} x_{i+1} \log_{x_i} n\right).$$

Now, choosing $t = \log \Delta - \log\log n + \log\log^a \log^\star n - 2$ times, for some $1 \le a \le \frac{\log\log n}{\log\log\log^\star n}$, and setting $x_{i+1} = 2x_i$ with $x_1 = 4$, we obtain a schedule of time complexity

$$O\Bigg(D + 4\log n + 8\log_4 n + \cdots + \frac{\Delta \log^a \log^\star n}{\log n(\log \Delta - \log\log n + \log\log^a \log^\star n - 2)} + \frac{\Delta \log n}{\log \Delta - \log\log n + \log\log^a \log^\star n}\Bigg),$$

which is $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n + \log\log^c \log^\star n})$ for some constant $c \ge 1$.

We have proved the following corollary.

Corollary 2 *In any graph G of $\Delta = \Omega(\log n)$, the gossiping task can be completed in time $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n + \log\log^c \log^\star n})$, for some constant $c \ge 1$.*

3 Broadcasting in Graphs with Known Topology

In this section we exploit the structure of the *SGST* for obtaining an algorithm that generates a schedule for completing the broadcasting task in a general known topology radio network in time $D + O(\frac{\log^3 n}{\log\log n})$.

In this case, a super-gathering spanning tree rooted at the *source node s* is used. The algorithm also uses the partition of the node set into the same transmission-sets F, S and *SS*. The broadcast message is now disseminated from the root towards the leaves of the tree. However, reversing the direction of the transmissions preserves the anti-collision capabilities of the structure. In particular, if in the gathering-stage of the gossiping scheme a node $v \in F$ could safely transmit to its parent, the same must hold true in the broadcast, for the transmission from *parent*(v) to v. Otherwise there would have to be a crossing edge causing collisions. Obviously such an edge cannot exist since it would have caused a collision in the gossiping scheme as well.

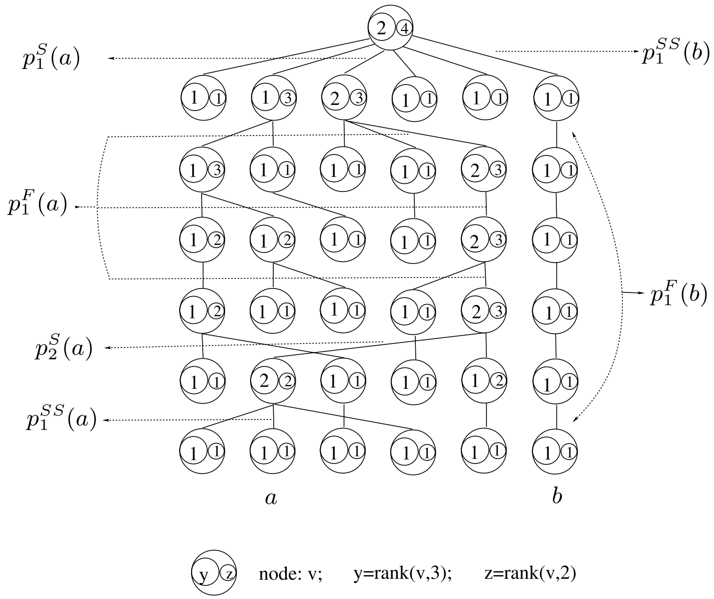

Fig. 5 An example of the path partition

For the sake of the analysis let us focus on a particular copy of the message that reaches a leaf a. Let $p(a)$ be the unique shortest path from the root s to the leaf a. Note that the message does not necessarily follow the path $p(a)$ and could actually even been delivered along non-shortest paths. Nonetheless the path $p(a)$ can be used to estimate the evolution of the message delivery from s to a. We can, e.g., measure the delay from the time the message is already available at some node v on the path $p(a)$ to the time the message has already reached the following node w on the path (though not necessarily via a transmission from v).

The path $p(a)$ can be though of as consisting of several segments

$$p(a) = \langle p_1^F(a), p_1^S(a), p_1^{SS}(a), p_2^F(a), p_2^S(a), p_2^{SS}(a), \dots, p_q^F(a), p_q^S(a), p_q^{SS}(a)\rangle,$$

where each $p_i^F(a)$ is a segment consisting of fast transmission edges (i.e., edges leading from $parent(v)$ to v of $\mathsf{rank}(parent(v), 2) = \mathsf{rank}(v, 2)$), each $p_i^S(a)$ is an edge (u, w) where u is a node on layer L_k for some k, w is a node on layer L_{k+1} and $\mathsf{rank}(u, 2) > \mathsf{rank}(w, 2)$ and $\mathsf{rank}(u, x) = \mathsf{rank}(w, x)$. We refer to such edges (u, w) as slow transmission edges. Further, each $p_i^{SS}(a)$ is an edge (y, z) where y is a node on layer L_k for some k and z is a node on layer L_{k+1} and $\mathsf{rank}(y, 2) > \mathsf{rank}(z, 2)$ and $\mathsf{rank}(y, x) > \mathsf{rank}(z, x)$. We refer to such edges (y, z) as super-slow transmission edges (see Fig. 5 for an example). (Note that some of the segments $p_i^F(a)$, $p_i^S(a)$ and $p_i^{SS}(a)$ may be empty.)

The progress of the message can be viewed as traversing the path $p(a)$ by alternating (flipping) among chains $p_i^F(a)$ of fast transmission edges, slow transmission steps over edges $p_i^S(a)$ and super-slow transmission edges $p_i^{SS}(a)$.

Next we describe the schedule governing these transmissions. During the broadcasting process the nodes in the tree use the following pattern of transmissions.

Consider a node v with $1 \leq \mathsf{rank}(v,2) \leq r^{[2]}_{\max}$ on BFS layer L_i with a child w of the same rank at the next BFS layer. Then v can perform a fast transmission to w in a time step t satisfying $t \equiv i + 9j \bmod 9r^{[2]}_{\max}$, where $j = \mathsf{rank}(v,2)$. The slow transmissions at the BFS layer L_i are performed in the time steps t satisfying $t \equiv i + 3 \bmod 9$. The super-slow transmissions at the BFS layer L_i are performed in the time steps t satisfying $t \equiv i + 6 \bmod 9$. This way, the fast, the slow and the super-slow transmissions at any BFS layer are separated by three units of time. Thus, there are no collisions between the fast, the slow and super-slow transmissions at the same BFS layer. Moreover, there cannot be conflicts between transmissions coming from different BFS layers either. In fact, at any time step, transmissions are performed on BFS layers at distances that are multiples of 3 apart.

When the message arrives at the first node v of a fast segment $p_i^F(a)$ of the route (with a particular rank), it might wait for as many as $9r^{[2]}_{\max} = O(\log n)$ time steps before being transmitted to the next BFS layer. However, it will then be forwarded through the fast segment $p_i^F(a)$ without further delays.

Once reaching the end node u of the fast segment $p_i^F(a)$, the message has to be transmitted from some node on u's BFS layer to the next node w on $p(a)$, which has $\mathsf{rank}(u,2) > \mathsf{rank}(w,2)$ and $\mathsf{rank}(u,x) = \mathsf{rank}(w,x)$, using a slow transmissions mechanism. For slow transmissions, the algorithm uses the x transmissions to progress distance one on $p(a)$ due to the property of the *SGST*.

Once reaching the end node y of the slow segment $p_i^S(a)$, the message has to be transmitted from some node on y's BFS layer to the next node z on $p(a)$, which has $\mathsf{rank}(y,2) > \mathsf{rank}(z,2)$ and $\mathsf{rank}(y,x) > \mathsf{rank}(z,x)$, using a super-slow transmissions mechanism. For super-slow transmissions, the algorithm uses the $O(\log^2 n)$ transmission Procedure CW proposed by Chlamtac and Weinstein in [3]. Procedure CW allows to move uniform information from one partition of a bipartite graph of size n (here, an entire BFS layer L_j of the tree) to the other (here, the next layer L_{j+1}) in time $O(\log^2 n)$. The super-slow transmission mechanism based on Procedure CW is run repeatedly in a periodic manner at every BFS layer of the tree. In particular, at any BFS layer, the steps of the super-slow transmission procedure CW are performed in every 9th step of the broadcasting schedule.

Hence, suppose the broadcast message traversing towards any destination a in the tree has reached a node y of BFS layer L_j on its path $p(a)$, such that the next edge (y,z) on the path is a super-slow transmission edge. It is possible that neither y nor any other neighbour of z on BFS layer L_j participates in the current activation of procedure CW on L_j (possibly because neither of those nodes had the message at the last time the procedure was activated). Nevertheless, y will participate in the next activation of procedure CW on BFS layer L_j, which will be started within at most $O(\log^2 n)$ time (namely, the time required for the current activation to terminate). Moreover, it is guaranteed that by the time that the procedure CW terminates, z will have the message (although it may get it from any of its neighbours in L_j, and not necessarily directly from y). Hence this entire stage can be thought of as a super-slow transmission operation on the edge (y,z), taking a total of at most $O(\log^2 n)$ time steps.

By virtue of the above observations we can bound the total time required for the broadcast message to reach a leaf a as follows. Let D_i, for $1 \le i \le r^{[2]}_{\max}$, denote the length of $p^F(a)$, the ith fast segment of the route $p(a)$ used by the broadcast message that has reached a. Thus the time required to communicate a is bounded by $O(\log n) + D_1 + \cdots + O(\log n) + D_{r^{[2]}_{\max}} \le D + O(\log^2 n)$ for the fast transmissions plus $r^{[2]}_{\max} \cdot O(x) = O(x \log n)$ for the slow transmissions and $\log_x n \cdot O(\log^2 n) = O(\frac{\log^3 n}{\log x})$ for the super-slow transmissions, yielding a total of $D + O(x \cdot \log n + \frac{\log^3 n}{\log x})$. The following theorem summarizes our findings.

Theorem 6 *There exists a deterministic polynomial time algorithm that constructs, for any n node radio network of diameter D, a broadcasting schedule of length $D + O(x \cdot \log n + \frac{\log^3 n}{\log x})$, where $2 \le x \le \log^2 n$. In particular, by setting $x = \log n$, we obtain the bound $D + O(\frac{\log^3 n}{\log\log n})$.*

4 Conclusion

We have proposed a new efficient (polynomial time) construction of a deterministic schedule that performs the gossiping task in radio networks in time $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$. The solution is based on the new concept of a super-gathering spanning tree. The new gossiping schedule is almost optimal since there exists a radio network topology, specifically a Δ-regular tree, in which the radio gossiping cannot be completed in less than $\Omega(D + \frac{\Delta \log n}{\log \Delta})$ units of time.

We also showed how the structure of the *SGST* can be used to obtain a schedule for completing the broadcasting task in a general known topology radio network in time $D + O(\frac{\log^3 n}{\log\log n})$. The interest in this result rely on the coefficient one of the D terms. Because of this, our suboptimal scheme compares favorably with the asymptotically optimal $O(D + \log^2 n)$ of [10], for instances with large D. The evident open problem regarding broadcast is then whether there exists a deterministic broadcast schedule of time $D + O(\log^2 n)$ for every n-node graph G of diameter D.

References

1. Alon, N., Bar-Noy, A., Linial, N., Peleg, D.: A lower bound for radio broadcast. J. Comput. Syst. Sci. **43**, 290–298 (1991)
2. Chlamtac, I., Kutten, S.: On broadcasting in radio networks-problem analysis and protocol design. IEEE Trans. Commun. **33**, 1240–1246 (1985)
3. Chlamtac, I., Weinstein, O.: The wave expansion approach to broadcasting in multihop radio networks. IEEE Trans. Commun. **39**, 426–433 (1991)
4. Diks, K., Kranakis, E., Krizanc, D., Pelc, A.: The impact of information on broadcasting time in linear radio networks. Theor. Comput. Sci. **287**, 449–471 (2002)
5. Elkin, M., Kortsarz, G.: Improved broadcast schedule for radio networks. In: Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, pp. 222–231 (2005)
6. Gaber, I., Mansour, Y.: Broadcast in radio networks. In: Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms, pp. 577–585 (1995)

7. Gąsieniec, L., Peleg, D., Xin, Q.: Faster communication in known topology radio networks. In: Proceedings of the 24th Annual ACM SIGACT-SIGOPS PODC, pp. 129–137 (2005)
8. Gąsieniec, L., Potapov, I.: Gossiping with unit messages in known radio networks. In: Proceedings of the 2nd IFIP International Conference on Theoretical Computer Science, pp. 193–205 (2002)
9. Gąsieniec, L., Potapov, I., Xin, Q.: Efficient gossiping in known radio networks. In: Proceedings of the 11th SIROCCO. Lecture Notes in Computer Science, vol. 3104, pp. 173–184. Springer, Berlin (2004)
10. Kowalski, D., Pelc, A.: Optimal deterministic broadcasting in known topology radio networks. Distributed Comput. **19**(3), 185–195 (2007)
11. Manne, F., Wang, S., Xin, Q.: Faster radio broadcast in planar graphs. In: Proceedings of the 4th Conference on Wireless on Demand Network Systems and Services. pp. 9–13. IEEE Press (2007)
12. Manne, F., Xin, Q.: Optimal gossiping with unit size messages in known topology radio networks. In: Proceedings of the 3rd Workshop on Combinatorial and Algorithmic Aspects of Networking. Lecture Notes in Computer Science, vol. 4235, pp. 125–134. Springer, Berlin (2006)
13. Sen, A., Huson, M.L.: A new model for scheduling packet radio networks. In: Proceedings of the 15th Joint Conference of IEEE Computer and Communication Societies, pp. 1116–1124 (1996)
14. Strahler, A.N.: Hypsometric (area-altitude) analysis of erosional topology. Bull. Geol. Soc. Am. **63**, 117–1142 (1952)
15. Viennot, X.G.: A Strahler bijection between Dyck paths and planar trees. Discrete Math. **246**, 317–329 (2002)